DBT Learning Notes – Data Build Tool

What is it? - A transformation workflow.

Why use it?

* Benefits with SE best practices like version control, modularity, portability, CI/CD and documentation.
* Dbt helps taking care of the materialization.
* Reduce the time queries take – leveraging **metadata** to find long-running models and use **incremental models**.
* DRYer code by macros, hooks and package management (what are these three???).

Important Features:

* SQL files can contain Jinja, a lightweight templating language. Using Jinja in SQL provides a way to use control structures in your queries. For example, if statements and for loops. It also enables repeated SQL to be shared through macros.
* Macros?
* Ref function?
* Auto-generate documentation
* Auto testing
* Package management – allow you to use and publish repositories so others can re-use them.
* Seed file?
* Snapshot?

Two ways to use dbt:

* dbt Cloud – managed by dbt team for infrastructure, making your life easier, but $ involved.
* dbt Core – open-source tool, manually setup and locally maintain it. Installed through command line.

My own practice:

I tried integrating one dbt initialization with my Snowflake trial account in my exiting project (streaming\_data\_lakehouse\_lab), the process is easy to follow. Some of the critical points include:

1. Other than using pip to install dbt-core, you will also need to install the adapter for the database you are trying to connect to.
   1. Here is the snowflake reference: <https://docs.getdbt.com/docs/core/connect-data-platform/snowflake-setup>
   2. <https://docs.getdbt.com/docs/core/pip-install>
2. When you run the dbt init command for the first time, its interactive cli will ask your snowflake connection information.
   1. When providing account name, you just need the <account-name> part in “<account-name>.snowflakecomputing.com”
3. After the dbt init commands completes, there will be a profile.yml file created under your user directory, in which the database connection info is stored. You can update this file later to change targets.
4. Meanwhile, there will be a dbt\_project.yml file created in your project folder. This file is critical to setup the project.
5. It seems like the table/view created in target DBMS has the same name as the script name.
   1. In the following experiments, I need to understand how we can assert the database and schema of the target. I don’t think we will need to create multiple dbt projects to perform the operations on each database.
6. Packages - Following ChatGPT, I was trying to use a dbt macro to re-create the schema.yml file. Here are some new learnings.
   1. Command I was trying to run: dbt run-operation generate\_model\_yaml --args '{"model\_name": "customers"}'
   2. Faced error that no package – then realized that to run a macro I need to install packages.
   3. Command: dbt deps (To test whether I have the package)
   4. Added info in yml file: A screenshot of a computer

      AI-generated content may be incorrect.
   5. Rerun Command: dbt deps – this will install that dbt-labs/codegen
   6. Later found that the command ChatGPT suggested was out-of-date, so checking: <https://github.com/dbt-labs/dbt-codegen/tree/0.13.1/#generate_model_yaml-source> to find out the right arguments to use.
      1. <https://hub.getdbt.com/dbt-labs/codegen/latest/>
7. <https://docs.getdbt.com/guides/bigquery?step=14> 🡪 Followed this tutorial but using dbt cli to test with creating tables/views in BigQuery.